

**MASARYKOVA UNIVERZITA**  
**PŘÍRODOVĚDECKÁ FAKULTA**  
ÚSTAV TEORETICKÉ FYZIKY A ASTROFYZIKY

# **Bakalářská práce**

**BRNO 2021**

**KRISTÝNA JANOUŠKOVÁ**

**M A S A R Y K O V A**  
**U N I V E R Z I T A**  
**PŘÍRODOVĚDECKÁ FAKULTA**  
ÚSTAV TEORETICKÉ FYZIKY A ASTROFYZIKY

---

# **Podrobné rádiové studium NGC 2329**

Bakalářská práce

**Kristýna Janoušková**

**Vedoucí práce: Mgr. Filip Hroch, Ph.D.    Brno 2021**

# Bibliografický záznam

<b>Autor:</b>	Kristýna Janoušková Přírodovědecká fakulta, Masarykova univerzita Ústav teoretické fyziky a astrofyziky
<b>Název práce:</b>	Podrobné rádiové studium NGC 2329
<b>Studijní program:</b>	Fyzika
<b>Studijní obor:</b>	Astrofyzika
<b>Vedoucí práce:</b>	Mgr. Filip Hroch, Ph.D.
<b>Akademický rok:</b>	2020/2021
<b>Počet stran:</b>	viii + 35
<b>Klíčová slova:</b>	CASA software; radiová galaxie; ngc2329; spektrální index; VLA; interferometrie

# Bibliographic Entry

**Author:** Kristýna Janoušková  
Faculty of Science, Masaryk University  
Department of theoretical physics and astrophysics

**Title of Thesis:** A comprehensive radio study of NGC 2329

**Degree Programme:** Physics

**Field of Study:** Astrophysics

**Supervisor:** Mgr. Filip Hroch, Ph.D.

**Academic Year:** 2020/2021

**Number of Pages:** viii + 35

**Keywords:** CASA software; radio galaxy; ngc2329; spectral index; VLA; interferometry

# Abstrakt

Tato bakalářská práce se zabývá zpracováním radiových dat galaxie NGC 2329 napozorovaných na observatoři VLA pomocí softwaru CASA (Common Astronomy Software Applications). V teoretické části představujeme část elektromagnetického spektra patřící do radiového oboru, dále popisujeme rozdíl mezi galaxiemi typu Fanaroff-Riley I (FRI) a Fanaroff-Riley II (FRII), a přinášíme základní informace o astronomickém interferometru a samotné observatoři VLA v Novém Mexiku. V praktické části se věnujeme kalibraci napozorovaných dat a self-kalibraci. V závěru práce se nachází spectral index mapa a popis její interpretace.

# Abstract

This bachelor thesis focuses on processing of the radio data of the galaxy NGC 2329 observed by the VLA, using CASA (Common Astronomy Software Applications) software. In the theoretical part we introduced part of the electromagnetic spectrum belonging to radio range, then we describe the difference between the Fanaroff-Riley I (FRI) and the Fanaroff-Riley II (FRII) type galaxies and then we bring basic information about astronomical interferometer and VLA observatory in New Mexico. In the processing part we focus on the calibration of the observed data and the self-calibration. In the end, the spectral index map is created and its interpretation is explained.

ZADÁNÍ  
BAKALÁŘSKÉ PRÁCE

Akademický rok: 2020/2021

Ústav:	Ústav teoretické fyziky a astrofyziky
Studentka:	Kristýna Janoušková
Program:	Fyzika
Obor:	Astrofyzika

Ředitel ústavu PŘF MU Vám ve smyslu Studijního a zkušebního řádu MU určuje bakalářskou práci s názvem:

Název práce:	Podrobné rádiové studium NGC 2329
Název práce anglicky:	A comprehensive radio study of NGC 2329
Jazyk závěrečné práce:	angličtina

**Oficiální zadání:**

Galaxies with active nuclei are very interesting objects since their discovery. This complicated physical phenomenon can demonstrate its power, and mysteries, by several ways. The most prominent is an unusual luminosity over wide parts of spectra; one is powered by black holes sitting in their cores.

Many of them are also active in radio bands. Their images reveal complex structures, commonly varying, in different angular resolutions and distances. There are also extensive radio lobes with dimensions highly exceeding sizes of their host galaxies; tails and jets are sources of the lobes.

The galaxy NGC 2329 with active nucleus is a relatively unknown, rarely observed object, while it is relatively bright, and not so far, as well as a member of a galaxy group. Some recent, very accurate, data are available for the galaxy.

This thesis will be focused on the radio study of the jets and lobes of the galaxy NGC 2329. Observed data will be processed by using CASA software, together with an advanced image processing, mathematical statistical methods in mind, as well as with deep astro-physical background.

Vedoucí práce:	Mgr. Filip Hroch, Ph.D.
Konzultant:	Mgr. Romana Grossová
Datum zadání práce:	24. 11. 2020
V Brně dne:	14. 6. 2021

Zadání bylo schváleno prostřednictvím IS MU.

Kristýna Janoušková, 21. 1. 2021

Mgr. Filip Hroch, Ph.D., 26. 1. 2021

Mgr. Michael Krbek, Ph.D., 1. 2. 2021

# Poděkování

Na tomto místě bych chtěla poděkovat svému vedoucímu doktoru Filipu Hrochovi za rady a nabízené dýchánky a své konzultantce Romaně Grossové, které měla dostatek trpělivosti odepisovat mi na každou otázku hned několikrát.

# Prohlášení

Prohlašuji, že jsem svoji bakalářskou práci vypracovala samostatně pod vedením vedoucího práce s využitím informačních zdrojů, které jsou v práci citovány.

Brno 14. června 2021

.....  
Kristýna Janoušková

# Contents

<b>Acronyms used</b> .....	<b>viii</b>
<b>Introduction</b> .....	<b>1</b>
<b>Chapter 1. The electromagnetic spectrum</b> .....	<b>2</b>
1.1 Radio bands .....	2
1.2 Synchrotron radiation .....	3
<b>Chapter 2. Radio interferometry</b> .....	<b>5</b>
2.1 Basics .....	5
2.2 VLA .....	6
<b>Chapter 3. Radio Galaxies</b> .....	<b>8</b>
3.1 FR I .....	8
3.2 FR II .....	8
3.3 The radio source NGC 2329 .....	9
<b>Chapter 4. Data processing</b> .....	<b>10</b>
4.1 Introducing the data .....	10
4.2 Obtaining the data .....	10
4.3 Calibration .....	11
4.3.1 Examining the data .....	11
4.3.2 Flagging .....	13
4.3.3 The calibration .....	15
4.4 Imaging .....	23
4.5 Spectral index map .....	27
<b>Chapter 5. Results</b> .....	<b>30</b>
<b>Conclusion</b> .....	<b>32</b>
<b>Bibliography</b> .....	<b>34</b>



# Acronyms used

Here we present all acronyms used in this text.

AGN	Active galactic nuclei
CMB	Cosmic microwave background
EM	electromagnetic
FOV	Field of view
FWHM	full width at half maximum
MS	measurement set
PSF	point spread function
RMS	root mean square
SNR	signal-to-noise ratio
WAT	Wide angle tailed

# Introduction

In the early 1930s, Karl Guthe Jansky, an employee of Bell Telephone Laboratories in New Jersey, built a directional antenna to identify sources of interference spoiling telephone communications. He discovered that thunderstorms distorted the transmission, but there was another source he couldn't figure out. His first thought was that it could be the Sun but after months of study he discovered that the signal was repeating every  $23^{\text{h}} 56^{\text{min}}$  and thus it must be some astronomical object. By comparing the position of the brightest point of the signal with astronomical optical maps, he concluded that the radiation comes from the constellation of Sagittarius ([Jansky 1933](#)), which Harlow Shapley established as the center of our Galaxy ([Shapley 1918](#)). Although Karl Jansky wanted to continue investigating the radiation from the centre of the Milky Way, the Bell Laboratories reassigned him to another project.

This discovery is now seen as the birth of radio astronomy. Astronomers yet remained unaware of this event, at least for the next decade. Its significance became apparent with the detection of the 21-cm hydrogen line, the quasars, and the cosmic microwave background (CMB).

From these days, many other discoveries were made. Radio astronomy has become an integral part of astronomy along with visible, infrared, X-ray astronomy, etc.

# Chapter 1

## The electromagnetic spectrum

The electromagnetic spectrum is divided into several ranges; each of these ranges is defined by the interval of the wavelengths (and thus frequencies). These ranges are gamma rays, X-rays, ultraviolet, visible, infrared, microwaves, and radio waves.

This thesis will mainly focus on radio waves, whose interval of frequencies is 3 MHz – 300 GHz (so the wavelengths vary from 100 m to  $10^{-3}$  m).

### 1.1 Radio bands

Radio window is region of the radio frequencies that passes through the Earth's atmosphere without any significant reflection or attenuation. It allows the radio telescopes on the Earth's surface to collect the radiation in this frequency range.

In the figure 1.1, one can see that the ionosphere limits the radio window at wavelengths greater than few meters and that the atmosphere absorption becomes significant in the sub-millimeter range. That is why we need to send observatories detecting those wavelength intervals to the orbit.

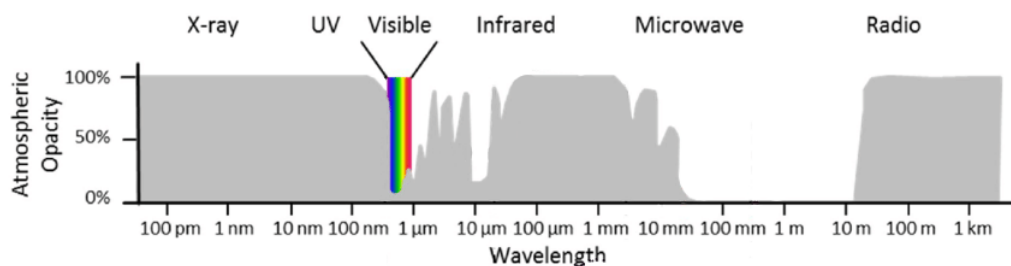


Figure 1.1: The atmospheric windows across the electromagnetic spectrum.

We further divide the radio range into several bands of frequencies. In the following chart, there are the most commonly used ones in astronomy:

Table 1.1: Overview of radio bands

Radio bands		
Frequency [GHz]	Wavelength [cm]	Band designation
1 – 2	30 – 15	L band
2 – 4	15 – 7.5	S band
4 – 8	7.5 – 3.75	C band
8 – 12	3.75 – 2.50	X band
12 – 18	2.50 – 1.67	Ku (U) band
18 – 27	1.67 – 1.11	K band
27 – 40	1.11 – 0.75	Ka band

## 1.2 Synchrotron radiation

Synchrotron radiation occurs when the charged particle (in our case, the electron but could be a proton, ion, etc.) is accelerated radially, e.g., by the magnetic field. It is non-thermal process, responsible for the observed radio continuum emission.

The process is similar to the cyclotron emission; the difference between these two is that the particles causing the synchrotron radiation have ultrarelativistic velocities (speed close to that of light) while the cyclotron particles don't.

When the electron encounters a strong magnetic field, it is radially accelerated, causing the electron to spiral around the magnetic field and radiate the photons. Due to the relativistic nature of the electrons the radiation is visible only for a short period for us (see in the figures 1.2 and 1.3).

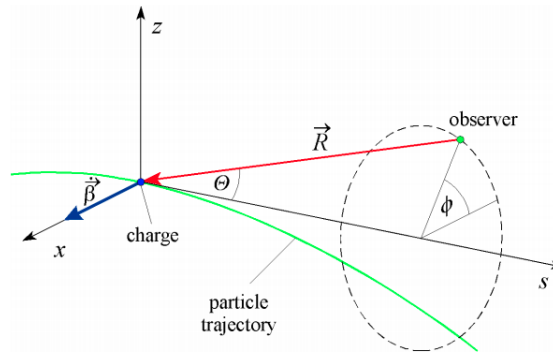


Figure 1.2: Coordinate system of the moving electron. Here  $\Theta$  is the opening angle depending on the electron's velocity by relation  $\Theta = 1/\gamma$  and  $\Phi$  is the polar angle dependent on the observer's position. (Wille 2013)

In the picture 1.3 there are four examples of the synchrotron radiation patterns for different  $\beta$  ( $\beta = v/c$ ) where  $v$  is the speed of the particle and  $c$  is the speed of light. The equation standing behind these patterns is the radiative power  $P$  per solid angle  $\Omega$  (Wiedemann 2019):

$$\frac{dP}{d\Omega} = \frac{1}{c^3 \mu_0} \frac{e^2}{(4\pi\epsilon_0)^2} \frac{\beta^4}{\rho^2} \frac{(1 - \beta \cos^2 \Theta) + (\beta^2 - 1) \sin^2 \Theta \cos^2 \phi}{(1 - \beta \cos \Theta)^5}. \quad (1.1)$$

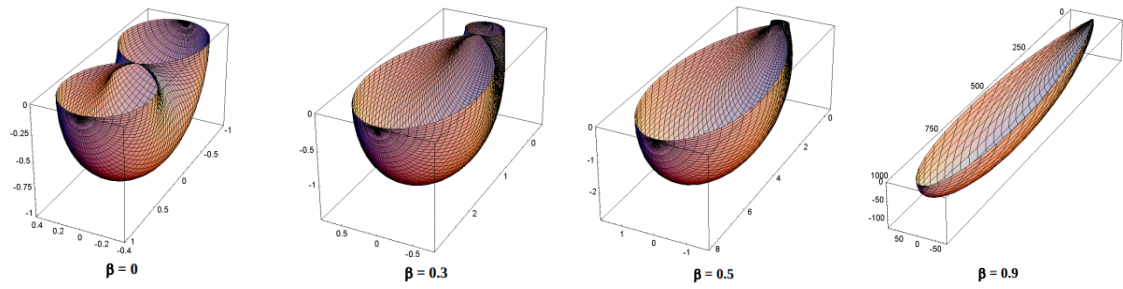


Figure 1.3: Radiation patterns for different  $\beta$  velocities (Wille 2013).

The spectrum of the single electron and from multiple electrons can be seen in the figure 1.4. The final spectrum from numerous electrons is the superposition of the spectra from the single electrons (as seen in the figure) and forms so – called power law shape. When there is continuous supply of new electrons, the spectrum is flat (slope of the power law varies between  $-0.5$  and  $0$ ) while when the slope is higher than  $0$ , the spectrum steepens. Steeper spectrum shows up in the radio galaxies' lobes for higher frequencies (Saikia et al. 2007), where electrons loose their energy with time.

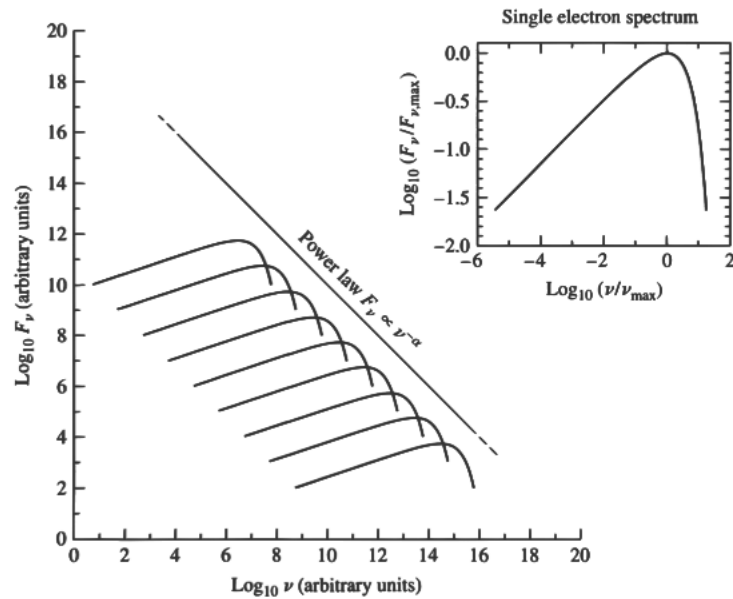


Figure 1.4: Spectrum of the synchrotron radiation caused by the single electron (right) and power law shape of the spectrum from multiple electrons (left) (Ertley 2014).

# Chapter 2

## Radio interferometry

### 2.1 Basics

Radio interferometer is an array consisting of two or more radio telescopes that work together to get higher angular resolution. The reason for using interferometers is that the limit of the resolution for single radio telescope is given by equation (2.1) (Rayleigh 1879):

$$\Theta \approx \frac{1.22\lambda}{D}, \quad (2.1)$$

where  $\lambda$  is the observed wavelength and  $D$  is the diameter of the antenna, and the angular resolution for the interferometer is

$$\Theta \approx \frac{\lambda}{b} \text{ [radian]}, \quad (2.2)$$

where  $b$  is the longest distance of the two single dish used in the interferometer (in radio astronomy jargon this distance is called the *baseline* and the resolution the *primary beam*).

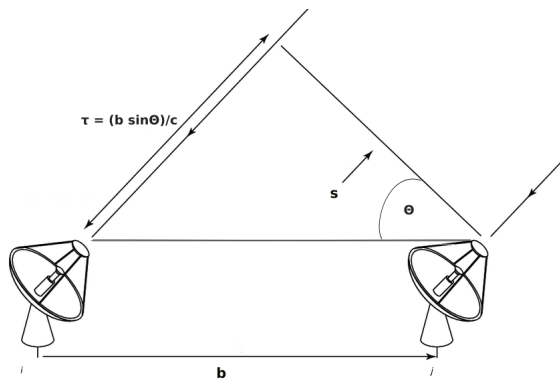


Figure 2.1: Scheme of the two-antenna interferometer. Here the  $\vec{b}$  is the baseline,  $\vec{s}$  the direction to the observed source and the  $\tau$  is the signal delay.

Another commonly used term is the *visibility*. The complex visibility function is the two dimensional Fourier transform of the emission on the sky. The real part of the visibility refers to the amplitude and the complex part stands for the phase.

## 2.2 VLA

VLA, or the Very Large Array, is the most productive centimeter – wavelength radio astronomy observatory. It is situated in the Plains of San Agustin (a flat part of a desert ringed by the mountains) in New Mexico, 80 kilometers northwest of Socorro, and it stands at an altitude of 2124 meters.



Figure 2.2: The iconic 'Y' shape of the VLA array in the D-configuration.

It consists of 28 identical paraboloids 25-meters in diameter. Twenty-seven of them are in operation at any given time, and the remaining one is a spare. The individual telescopes, deployed in a Y-shape, are connected by optical fibers. Each antenna has ten feed-horn receivers.

Table 2.1: Ten radio receivers used in VLA dishes.

Frequency	Band designation	Frequency	Band designation
74 MHz	Band 4	8.4 GHz	X band
327 MHz	P Band	15 GHz	Ku Band
1.4 GHz	L Band	22 GHz	K Band
3 GHz	S Band	33 GHz	Ka band
5 GHz	C Band	43 GHz	Q Band

Each one is cryogenically cooled to keep their materials and electronics from giving off any radio signals of their own that would swamp the incoming radio signal from the source.

And why that iconic "Y" shape? Each of the "Y" branches is made up of a rail track and nine antennas. Four times a year, a rail transporter picks up one antenna at a time and drives it farther from the center (so the whole array lengthens from 1.06 km to 37 km). That means, four times a year, there is a change in the resolution of the observatory.

With four of the possible layouts of the antennas, there come four configurations. From the largest to the most compact, these are A, B, C, and D. Each configuration has its own field of view and angular resolution.

The following figure shows the differences in observing the same object with different array diameters.



Figure 2.3: Different resolutions for every configuration of the VLA array. Credits: NRAO



# Chapter 3

## Radio Galaxies

Radio galaxies are a type of active galactic nuclei (AGN) that emit much more radio emission (luminosities up to  $10^{39}$  W) than visible.

Mostly these are large elliptical galaxies that show enormous regions of radio emission outside the visible extent of the galaxy. These so-called radio lobes can be hundreds of kilo light years wide. Radio lobes usually come in pairs and do not emit visible light.

Fanaroff and Riley classified the radio galaxies into two main types ([Fanaroff & Riley 1974](#)) based on the galaxy's morphology. Those having edge-darkened morphology bear the designation FR I and those with edge-brightened morphology are FR II.

### 3.1 FR I

This type becomes fainter with increasing distance from the source; that's why they were called edge-darkened sources. The jets are nicely visible, but the lobes slowly fade out.

A typical FR I galaxy is shown below:



Figure 3.1: The radio source 3C 449 as a typical representative of the FRI type radio galaxy. ([Perley et al. 1979](#))

### 3.2 FR II

This class comprises radio sources with hotspots and bright lobes. Compared to FR I galaxies, these have a brighter core and jets (they are overall more luminous than FR I), but these features are much fainter in FR II relative to the lobes.

The first radio galaxy ever discovered, Cygnus A, is a typical representative of FR II-type. This object is still the brightest known radio galaxy.

Another object belonging to this group is 3C 47, shown in the figure 3.2.

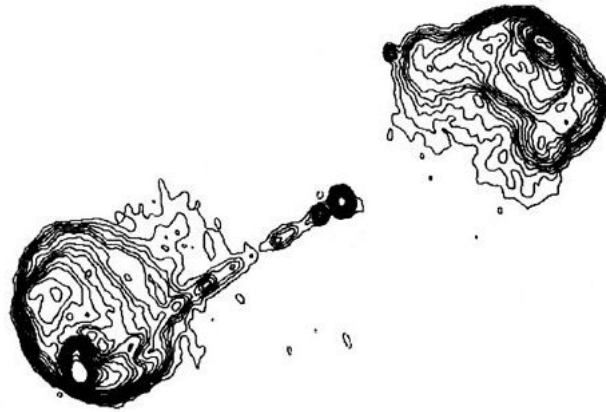


Figure 3.2: Quasar 3C 47. Typical FR II radio galaxy ([Bridle et al. 1994](#))

### 3.3 The radio source NGC 2329

NGC 2329 (RA  $07^{\text{h}} 09^{\text{m}} 08.1^{\text{s}}$ , Dec  $+48^{\circ} 36' 58''$ , J2000) is an elliptical galaxy of Hubble classification E/S0 discovered by William Herschel on 9<sup>th</sup> of February 1788. It is a member of galaxy cluster Abell 569 in the northern Lynx constellation. With the magnitude of 13.6, this object is the brightest one of this cluster, but it is still little explored. NGC 2329 ranks among the wide angle tailed (WAT) sources ([Feretti et al. 1985](#)). There are no more articles on this source and that makes it interesting.

It is also observed to be associated with an X-ray source.

Abell 569 consists of two main condensations separated about 1.5 Mpc in the distance of 270 millions lyr. Probably it belongs to Perseus supercluster.

# Chapter 4

## Data processing

### 4.1 Introducing the data

For this thesis, we had one Jansky VLA dataset of an estimated size 21.26 GB.

We present basic information about the data:

- 1.51GHz, L-band, A-configuration, observed with the Jansky VLA on 20 June 2015 by Norbert Werner under the project code VLA/15A-305. The observation was performed using 27 VLA antennas and lasted for 59 minutes. The bandpass (and flux) calibrator was observed at the beginning for 12 minutes, and then the target observation was divided into eight scans alternated by 0.5-minute observation of the phase calibrator. The band is divided into 16 spectral windows, each with 64 channels. The lowest spectral window has the frequency of 1008 GHz and the highest 1968 GHz.

### 4.2 Obtaining the data

The first step to start with calibration and imaging is to obtain the data. The easiest way to get them is to download them directly from the [NRAO data archive \(2020\)](#).

Here we can fill in the required parameters. Our dataset has the following requirements:

We wanted data from the Jansky VLA (data after 2011 VLA upgrade), so we checked the 'Jansky VLA' box; in the section 'Position Search,' we filled in the 'Target Name' as 'NGC 2329'. To ease searching for the best dataset, we chose the 'Telescope Config A' and 'Observing Bands' as 'L', but this was not mandatory. After setting all of these requirements, we submitted the query via 'Submit Query' and proceeded to the second part of obtaining the data.

After filling in the email address, it is necessary to choose the download data format as 'CASA MS' and to check 'Create tar file' as 'Create MS or SDM tar file' in the section 'Jansky VLA datasets'. At the bottom of the page, check the archive required file and confirm by hitting the 'Get my data' button.

It takes from few minutes to several hours to copy the archive data (depending on the size of the final dataset); after this time, email with the subject 'Archive Data Copied' with

the link to download the file will be received.

The last step to get the data is to open the terminal and type:

```
#in terminal

wget ftp://ftp.aoc.nrao.edu/e2earchive/15A-305.sb30903437.
eb30964532.57193.88261458333.ms.tar
```

## 4.3 Calibration

### 4.3.1 Examining the data

Before starting the calibration, we need some basic information about our data. To get acquainted with the observing conditions, it is essential to download the observers log from [Operator's log \(2020\)](#). We fill up the start and the end of the observation (in our case, both start and stop dates are 2015-06-20). Now select the correct code for the observation (15A-305) and click the *pdf*.

We know that some antennas are not correctly situated from the log; we can start data reduction. The task `listobs` gives a complete overview of the observation (times of single-target observations, list of fields (observed objects), and the information about antennas). Our downloaded measurement set's (*.ms*) title is too long, but there is the possibility to simplify it into *15A-305.ms*.

```
#in CASA

listobs(vis='15A-305.ms')
```

Parameter *vis* is the abbreviation for the 'visibility', and it expects input in the form of a string. Fill in the title of our measurement set with the *.ms* extension.

The output in the CASA logger will show the block of information about the time of the observation with scans and fields described; the next block is about the fields and the spectral windows, and then we have information about sources. The final section carries information about the VLA antennas, e.g., the ID, name, offset from the array center, etc.

Note that the antenna IDs do not correspond to actual antenna names! E.g., the name of the antenna with ID '4' is 'ea05'.

To get the sense of the array and to identify antennas, use the task `plotants`:

```
plotants(vis='15A-305.ms', graphs
figfile='plotants_15A-305_antenna_layout.png')
```

*figfile* parameter gives the name to the output file that will be generated. The layout during this observation for this observation is displayed in the figure 4.1.

Now that we have the initial information, we can start inspecting the data.

The task `plotms` helps with checking of the problematic data. The best way to examine bad data is to plot "amplitude vs UVdist" (figure 4.2).

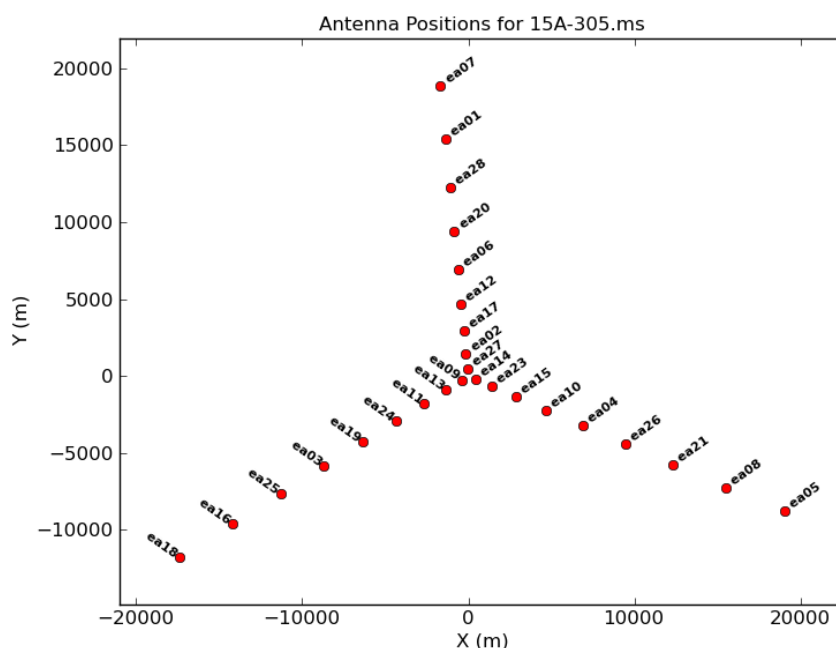


Figure 4.1: VLA antennas layout during the observation.

```
plotms(vis='15A-305.ms', xaxis='uvdist', yaxis='amp',
       coloraxis='baseline', plotfile='plotms_15A-305_
       ampsuvdist_All.png', showgui=False)
```

The default *x*- and *y*-axis parameters are the 'time' and 'amp', so we need to change them to produce the desired plot. By selecting the *coloraxis* as 'baseline', the resulting graph will look clearer. Because the Jansky VLA datasets are large, the loading process of the plots without specifying and averaging of the data took a huge amount of time. To reduce the time, we won't let the plotms GUI open, and we save the plot straight away in our working directory (*showgui=False*) instead.

From the graph that appeared in our working directory we see several peaks that are much higher than the majority. Those need to be flagged. From this file, however, we have no way of knowing what data are causing these peaks.

To examine which data are wrong, we need to create a plot using plotms GUI. To reduce the time of loading the chart, we split the data into separate fields (from the `listobs` task, we know that we have three of them – the target NGC2329 (field 2), the secondary phase calibrator J0710+4732 (field 1) and the primary flux calibrator 3C147 (field 0)).

```
plotms(vis='15A-305.ms', xaxis='uvdist', yaxis='amp',
       coloraxis='baseline', field='1')
```

The *field* parameter allows us to choose which of the observed objects we want to get acquainted with (4.2).

By filling up the *spw* box in the left panel from 0 to 15, check each spectral window.

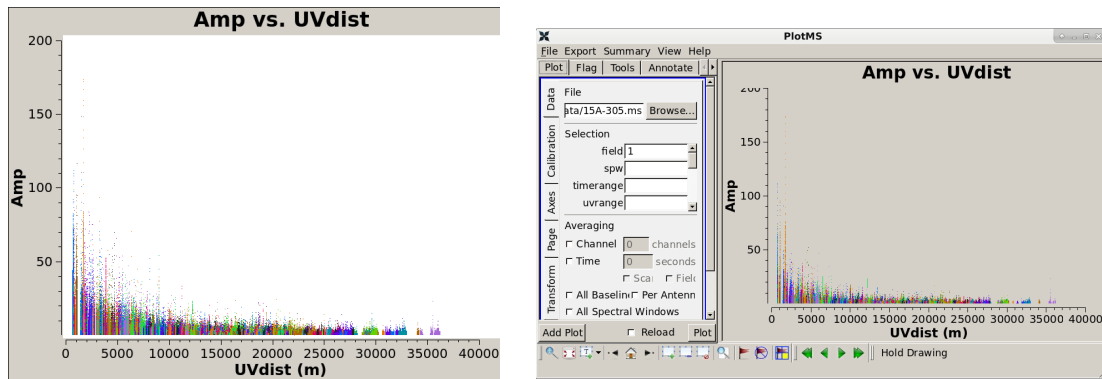


Figure 4.2: Amplitude versus UV distance plot saved without opening plotms GUI. We can see some peaks that need to be flagged (*left*) and amplitude versus UV distance plot for the field='1' and all spectral windows opened via plotms GUI (*right*).

Some would need to be flagged completely (the comparison of corrupted and smooth course of the graph are shown in the figure 4.3).

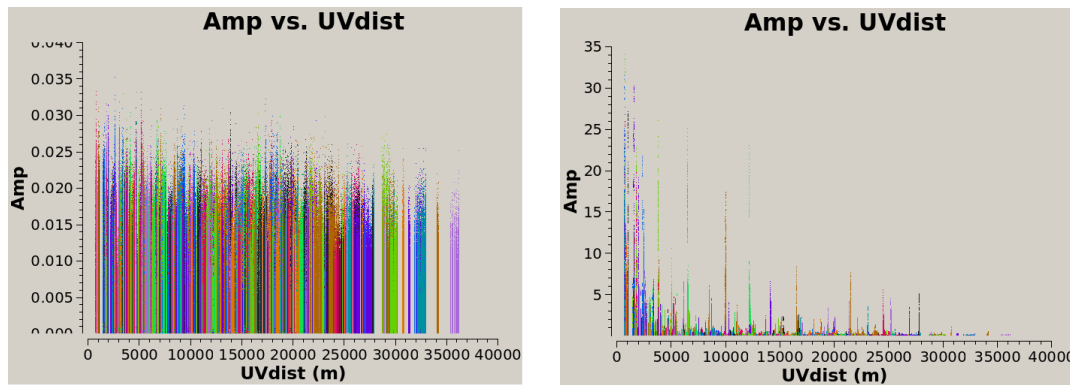


Figure 4.3: An example of an appropriate course of the plot (the peaks vary between the minimal range of amplitudes) – *left* and an example of corrupted spectral window – *right*. This one needs to be flagged completely (alongside the spw='3').

We need to determine which data are wrong and, therefore, inapplicable from the rest of the spectral windows (those that don't have any of the courses shown in the figure above). Mark some of the points from the highest peaks with the 'Mark Regions' button on the bottom of the GUI window and then click on 'Locate'. Results will be shown in the log message. From the outcome, we see that the baselines 'ea14&ea27', 'ea11&ea13', and 'ea23&ea27' are bad and need flagging.

This procedure should be done for every field.

### 4.3.2 Flagging

To flag the data means to mark them so they will not be further used.

As the first flag option we can choose *mode='fscrop'*. It is the autoflag algorithm that works with each field, spectral window, timerange, and baseline. This mode forms an

average spectrum by averaging of visibility amplitude along time dimension, calculates polynomial fit at the base of RFI spikes, and flags points deviating from the fit by some given value. Running this mode of flagging gives the best result when used on the beginning of the data processing.

From the `listobs` call, we see that scan 1's comment says 'system configuration', therefore it is necessary to flag this scan.

```
flagdata(vis='15A-305.ms', mode='tfcrop', datacolumn='data')

flagdata(vis='15A-305.ms', flagbackup=True, mode='manual', scan='1')
```

If `flagbackup` is set to `True`, `flagdata` saves a copy of the existing set of flags before entering new flags. One can choose to backup just the major flags, all of the flags or none of them. The `mode='manual'` allows us to select specific data to be edited.

Now is the time for the corrupted spectral windows. Spectral window '9' should be flagged together with spectral windows '3' and '8', since it is known to cause troubles.

```
flagdata(vis='15A-305.ms', flagbackup=True, mode='manual',
        spw='8; 9; 3')
```

Next are the improper baselines we have detected a few steps ago.

```
flagdata(vis='15A-305.ms', flagbackup=True, mode='manual',
        antenna='ea11&ea13; ea14&ea27; ea23&ea27')
```

In `antenna` parameter can be written just corrupted antenna or the whole baseline. For the baseline, there must be the '&' sign separating the two antennas.

The final flag will be done for the first 10 seconds of each scan.

```
flagdata(vis='15A-305.ms', mode='quack', quackinterval=10.0,
        quackmode='beg')
```

'*quack*' is another mode that applies the same edit to every scan and every baseline. The length of the time interval is set with the `quackinterval=10.0`; in this case, it takes 10 seconds on the beginning (`quackmode='beg'`) of every scan.

To be sure that we flagged everything needed, check again 'amp vs UVdist' plot.

The final helpful plot is a data stream of the `antenna2` for data versus `ea01`. This shows, assuming that `ea01` was in the entire observation, when various antennas dropped out.

```
plotms(vis='15A-305.ms', field='', correlation='RR,LL',
        timerange='', antenna='ea01', spw='*:31', xaxis='time',
        yaxis='antenna2', plotrange=[-1,-1,0,26], coloraxis='field')
```

By the `antenna` parameter we set the reference antenna for this plot. `Spw='*:31'` takes the channel number 31 from all of the spectral windows.

The resulting plot is displayed in the figure 4.4.

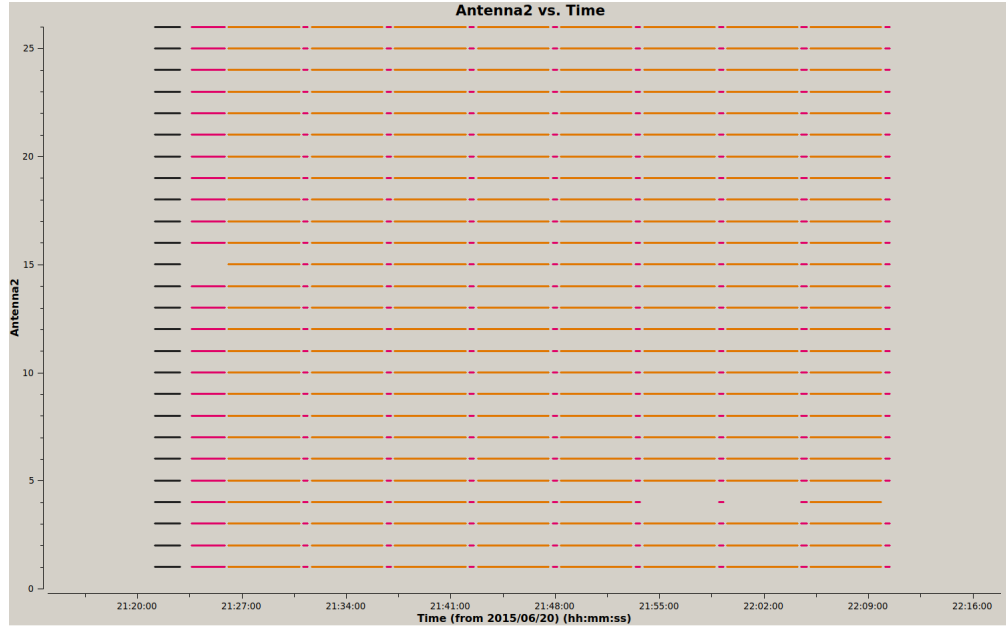


Figure 4.4: Datastream view of 15A-305.ms

We see that antennas ID 15 and 4 had some outages during the observation, so it is essential to keep this in mind and not select them as reference antennas in oncoming calibration.

### 4.3.3 The calibration

Now that the data are updated, we can start the calibration.

The observed visibility  $V'$  between two antennas  $(i, j)$  is a function of frequency  $\nu$  and depends on the spatial wave numbers  $\nu$  and  $u$ .

$$V'_{i,j}(u, \nu, \nu) = b_{i,j}(t)[B_i(\nu, t)B_j^*(\nu, t)]g_i(t)g_j(t)V_{i,j}(u, \nu, \nu)e^{i[\theta_i(t)-\theta_j(t)]} \quad (4.1)$$

$V_{i,j}$  is the true visibility;  $g_i$  and  $\theta_i$  are the portions of the complex gain. These portions are usually calculated separately (one represents the amplitude and the second the phase), so we showed them separately in this equation, too. They are dependent on the time (because of the temperature of the atmosphere, conditions etc.).  $B_i$  stands for the complex bandpass which is the response of the interferometer as a function of the frequency  $\nu$ . The final term  $b(t)$  could be important shortly after the configuration change of the array (NRAO 2021).

The first step is to find the baseline corrections. From the observers log, we know that some antennas are not in their correct positions. If the baselines are incorrect, the  $u$  and  $\nu$  will be calculated incorrectly and errors will occur in the images.

The calculations are inserted using task `gencal`:

```
gencal(vis='15A-305.ms', caltable='15A-305.antpos', caltype='antpos')
```

For correcting the antenna positions is used the `caltype='antpos'`.



The output is shown in the log and should contain information about antenna position corrections:

```
INFO Calibrator Initializing nominal selection to the whole MS.
INFO gencal Determine antenna position offsets from the baseline
      correction database
INFO gencal offsets for antenna ea01 : -0.00500    0.00600   -0.00700
INFO gencal offsets for antenna ea03 : -0.00700    0.00200    0.00000
INFO gencal offsets for antenna ea04 : -0.01400   -0.01000   -0.02000
INFO gencal offsets for antenna ea05 :  0.00000   -0.00200    0.02300
INFO gencal offsets for antenna ea06 : -0.00100    0.00000   -0.00500
INFO gencal offsets for antenna ea07 : -0.00200    0.00800   -0.00800
INFO gencal offsets for antenna ea08 :  0.00500   -0.00500    0.00700
INFO gencal offsets for antenna ea10 :  0.00300    0.00600    0.03200
INFO gencal offsets for antenna ea11 : -0.01000    0.03200    0.01600
INFO gencal offsets for antenna ea13 : -0.00100    0.00300    0.00900
INFO gencal offsets for antenna ea16 :  0.01200    0.00000   -0.02700
INFO gencal offsets for antenna ea17 :  0.01600    0.00800    0.02000
INFO gencal offsets for antenna ea18 :  0.01300    0.00800    0.00000
INFO gencal offsets for antenna ea19 :  0.01200   -0.01400   -0.00200
INFO gencal offsets for antenna ea20 : -0.00300    0.00300   -0.00500
INFO gencal offsets for antenna ea21 :  0.01200   -0.01800    0.01200
INFO gencal offsets for antenna ea25 :  0.00200    0.00200   -0.00900
INFO gencal offsets for antenna ea26 :  0.00000    0.02000    0.01500
INFO gencal offsets for antenna ea28 : -0.00300    0.00500   -0.00900
```

Next important step to calibrate the data is to provide a flux density value for the amplitude calibrator (3C147). First thing is to find the available calibrator models using task `setjy` and setting the parameter `listmodels = True`.

```
setjy(vis='15A-305.ms', listmodels=True)
```

This call will show all available calibrator models due to that `listmodels=True`:

```
No candidate modimages matching '*.im* *.mod*' found in .

Candidate modimages (*) in <your_path_to_CASA>/nrao/VLA/CalModels:
3C123_P.im 3C138_Q.im 3C147_K.im 3C147_X.im 3C286_P.im 3C380_P.im
3C48_Q.im
3C138_A.im 3C138_S.im 3C147_L.im 3C196_P.im 3C286_Q.im 3C48_A.im
3C48_S.im
3C138_C.im 3C138_U.im 3C147_P.im 3C286_A.im 3C286_S.im 3C48_C.im
3C48_U.im
3C138_K.im 3C138_X.im 3C147_Q.im 3C286_C.im 3C286_U.im 3C48_K.im
3C48_X.im
3C138_L.im 3C147_A.im 3C147_S.im 3C286_K.im 3C286_X.im 3C48_L.im
README
3C138_P.im 3C147_C.im 3C147_U.im 3C286_L.im 3C295_P.im 3C48_P.im
```

The flux density scale at the VLA is periodically revised, updated, or expanded. The default standard now is the 'Perley-Butler' database of the flux scale calibrators ([Perley & Butler 2017](#)). We need to choose our flux calibrator in the correct band ('3C147\_L.im').

```
setjy(vis='15A-305.ms', field='0', standard='Perley-Butler 2017',
      model='3C147_L.im', usescratch=False, scalebychan=True)
```

It is important to specify the *field*, because if not specified then all the sources will be assumed to have the same flux density. The *standard* = 'Perley-Butler 2017' parameter will use the latest information about the flux density scale for the calibrator, *model* = '3C147\_L.im' is the model chosen from the previous command. To save disk space, we will set the *usescratch* parameter as *False*. That means, the model visibilities will not be written to the MODEL\_DATA column. In this case, CASA will calculate and use the individual model visibilities when needed for calibration and for the plotms. Finally, the *scalebychan=True* allows the *setjy* task to determine a flux density value per channel instead of one value for the entire spectral window. This is useful for calibrators with intrinsic spectral index.

In CASA command line one should find output similar to this:

```
{'0': {'0': {'fluxd': array([ 28.78127098, 0. , 0. , 0. ])}},
'1': {'fluxd': array([ 27.5220089, 0. , 0. , 0. ])}},
'10': {'fluxd': array([ 19.7273922, 0. , 0. , 0. ])}},
'11': {'fluxd': array([ 19.12361336, 0. , 0. , 0. ])}},
'12': {'fluxd': array([ 18.55533791, 0. , 0. , 0. ])}},
'13': {'fluxd': array([ 18.01952553, 0. , 0. , 0. ])}},
'14': {'fluxd': array([ 17.51347733, 0. , 0. , 0. ])}},
'15': {'fluxd': array([ 17.0347805, 0. , 0. , 0. ])}},
'2': {'fluxd': array([ 26.36753845, 0. , 0. , 0. ])}},
'3': {'fluxd': array([ 25.30532074, 0. , 0. , 0. ])}},
'4': {'fluxd': array([ 24.32473373, 0. , 0. , 0. ])}},
'5': {'fluxd': array([ 23.41672325, 0. , 0. , 0. ])}},
'6': {'fluxd': array([ 22.57352066, 0. , 0. , 0. ])}},
'7': {'fluxd': array([ 21.78842926, 0. , 0. , 0. ])}},
'8': {'fluxd': array([ 21.05564308, 0. , 0. , 0. ])}},
'9': {'fluxd': array([ 20.37010765, 0. , 0. , 0. ])}},
'fieldName': '0'},
'format': "{field Id: {spw Id: {fluxd: [I,Q,U,V] in Jy}},
'fieldName':field name }}"}
```

The fundamental calibration of the data can be done using task *gaincal*. This command calculates the complex time-dependent gains for each antenna and spectral window and the results are determined from the ratio of the raw data divided by the model column. The *gaincal* parameter *gaintype* has many options from which we will use three of them in the rest of this section.

Initial phase calibration should be done first. Reason for this step is to average over the variations of phase with time in the bandpass to prevent de-correlation when vector averaging the data or computing the final bandpass. Amplitude variations with time have

only a little effect on the bandpass solutions so it is not necessary to solve it for.

```
gaincal(vis='15A-305.ms', caltable='15A-305.G0all', field='0,1',
        refant='ea09', spw='*:27~36', gaintype='G', calmode='p',
        solint='int', minsnr=5, gaintable=['15A-305.antpos'])
```

From this command we see that the final gain solutions will be stored in the '15A - 305.G0all' external table and it is set by the *caltable* parameter. Although the bandpass solution is calculated only for the bandpass calibrator, it is good idea to inspect all of the calibrators to prevent bad data. This is the first time of using the reference antenna. Generally, the antenna closest to the center without shadowing and without any observing outage is used as the reference antenna. For the phase calibration we will choose the central 10 % of the channels (so that a reasonable signal-to-noise ratio can be obtained) for each spectral window (this is ensured with the ' $*$ : 27 ~ 36'). From these the command will determine the gain corrections. Then, the *gaintype* computes the complex gain solutions, one per antenna per spectral window per polarization per solution interval. The 'G' type assumes the V Stokes is zero if not told otherwise (in case that the calibrator has significant circular polarization, a model incorporating polarization must be used).

```
plotms(vis='15A-305.G0all', xaxis='time', yaxis='phase', coloraxis=
        'corr', iteraxis='antenna', plotrange=[-1,-1,-180,180])
```

The task *plotms* is used to examine the solutions of the calibration table, too. The *vis='15A-305.G0all'* will show the newly created table for initial phase calibration with phase on the y-axis and time on the x-axis. The *coloraxis='corr'* will plot the points of parallel (RR and LL) correlations in different colors. *iteraxis='antenna'* allows us to see the solutions separately for each antenna (we can switch between them with the green arrow on the bottom of the window). The range we used is to restrict the area we're looking at.

In the plot shown in the figure 4.5, we see quite smooth variations with time. Antennas that have jumps in the variations need flagging. In our data, no antenna has phase jumps, so no flagging is needed.

Next task is *gaincal* again, but this time we will solve for bandpass calibrator (specifying by the *field* parameter) only. Then we can inspect our solutions again with the *plotms* as we did above. The calibration command for solving for the bandpass solution is shown below:

```
gaincal(vis='15A-305.ms', caltable='15A-305.G0', field='0',
        refant='ea09', spw='*:27~36', calmode='p', solint='int',
        minsnr=5, gaintable=['15A-305.antpos'])

plotms(vis='15A-305.G0', xaxis='time', yaxis='phase', coloraxis=
        'corr', field='0', iteraxis='antenna', plotrange=[-1,-1,-180,180],
        timerange='21:12:00~21:20:00')
```

Next stage of the bandpass calibration is the delay calibration. The *gaintype='K'* solves for simple-antenna delays to the reference antenna. It is recommended to cut off

approximately five channels on the edges of each spectral window as these have lower sensitivity (amplitude is dropping rapidly in these channels as shown in the figure 4.6). We must use the previously created calibration tables to calculate delays properly.

```
gaincal(vis='15A-305.ms', caltable='15A-305.K0', field='0',
        refant='ea09', spw='*:5~58', gaintype='K', solint='inf',
        combine='scan', minsnr=5, gaintable=['15A-305.antpos',
        '15A-305.G0'])

plotms(vis='15A-305.K0', xaxis='antenna1', yaxis='delay',
        coloraxis='baseline')
```

The figure 4.5 shows the delay solutions with no major problems detected.

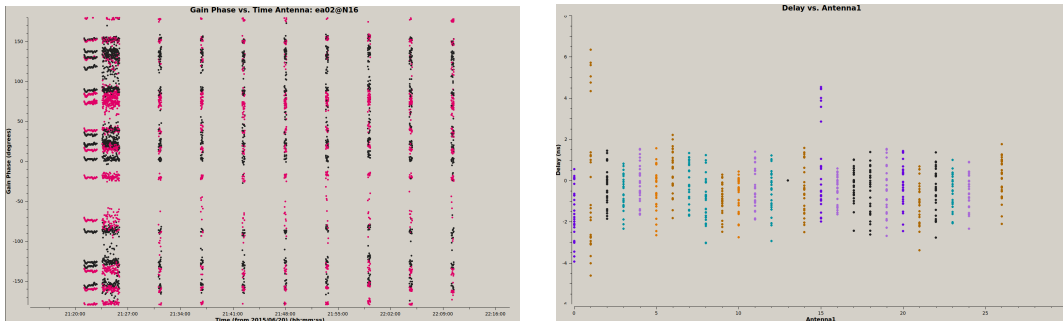


Figure 4.5: Solutions for the initial gain calibration. We can see smooth variations of the phases. Antennas with bigger deviations must be flagged (*left*) and solutions obtained from the gaincal calibration tables (*right*).

Finally, this step solves for the complex bandpass  $B_i$ . Bandpass solution is required to account for the gain variations with frequency that are a reflection of the different antenna bandpasses.

```
bandpass(vis='15A-305.ms', caltable='15A-305.B0', field='0',
        refant='ea09', combine='scan', solint='inf', bandtype='B',
        gaintable=['15A-305.antpos', '15A-305.G0', '15A-305.K0'])

plotms(vis='15A-305.B0', field='0', xaxis='chan', yaxis='amp',
        coloraxis='corr', iteraxis='antenna', gridrows=2, gridcols=2)

plotms(vis='15A-305.B0', field='0', xaxis='chan', yaxis='phase',
        coloraxis='corr', plotrange=[-1,-1,-180,180], iteraxis='antenna',
        gridrows=2, gridcols=2)
```

The solutions can be seen in the figures 4.6. Ideally, the 'phase vs chan' should have negligible variations around the zero value.

The final gaincal calibrations solve for the complex gain (the amplitude  $g_i$  and the phase  $\theta_i$  portions) for both calibrators. These will derive the corrections for the complex antenna gains.

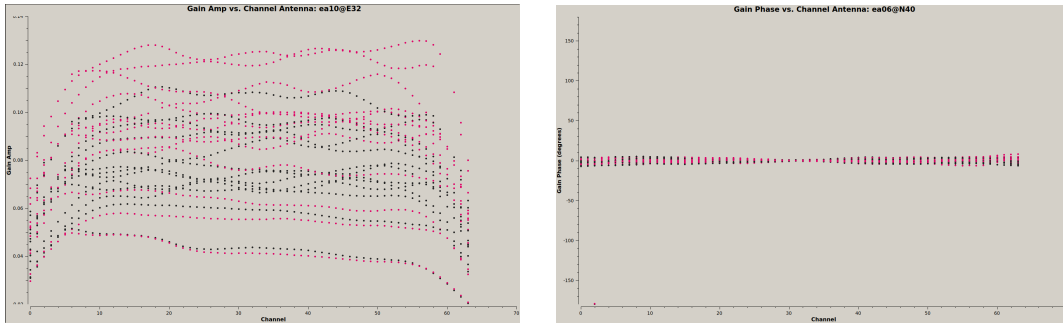


Figure 4.6: On the left there is a illustration of a dependence of bandpass amplitudes on the channel. We can see that the edge channels drop rapidly. On the right is bandpass solutions obtained running the task bandpass.

```
gaincal(vis='15A-305.ms', caltable='15A-305.G1', field='0', spw=
    '*:5~58', solint='inf', refant='ea09', gaintype='G', calmode='ap',
    solnorm=False, gaintable=['15A-305.antpos', '15A-305.K0',
    '15A-305.B0'], interp=['','','nearest'])

gaincal(vis='15A-305.ms', caltable='15A-305.G1', field='1', spw=
    '*:5~58', solint='inf', refant='ea09', gaintype='G', calmode='ap',
    gaintable=['15A-305.antpos', '15A-305.K0', '15A-305.B0'],
    append=True)

plotms(vis='15A-305.G1', xaxis='time', yaxis='phase', gridrows=1,
    gridcols=2, iteraxis='corr', coloraxis='baseline', plotfile=
    'plotms_ngc2329-G1-phase.png')

plotms(vis='15A-305.G1', xaxis='time', yaxis='amp', gridrows=1,
    gridcols=2, iteraxis='corr', coloraxis='baseline', plotfile=
    'plotms_ngc2329-G1-amp.png')
```

With the *gaintype*='G' we will find the calibrations separately for each spectral window and the *calmode*='ap' we ensure to have solution for the complex gain.

Last step of the calibration is scaling of the amplitude gains. We know the flux density of the primary calibrator but the model for the secondary calibrator was set to be a point source of 1 Jy (while the absolute flux density is unknown).

Finding the true flux density of the secondary calibrator is done by the *fluxscale* task which produces another calibration table containing calculated amplitude gains for the secondary calibrator.

```
myscale = fluxscale(vis='15A-305.ms',
                    caltable='15A-305.G1',
                    fluxtable='15A-305.fluxscale1',
                    reference='0',
                    transfer=['J0710+4732'],
                    incremental=False)
```

The field of the calibrator with the known flux is set by the *reference*='0' parameter while the field of the calibrator we want to determine the flux density of is given into the *transfer* parameter. The *incremental=False* makes a new output table replacing the old *caltable* generated few steps above.

Now that the calibrations are done and seem correct, we can apply all the calibrations to our MS for all of our fields. Do not do this until satisfied with all the plots.

The calibrations must be applied to the calibrators themselves first; the easiest approach is to run each *applycal* task for each calibrator separately.

```
applycal(vis='15A-305.ms',
        field='0',
        gaintable=['15A-305.antpos', '15A-305.fluxscale1',
                  '15A-305.K0', '15A-305.B0'],
        gainfield=['', '0', '', ''],
        interp=['', 'nearest', '', ''],
        calwt=False)

applycal(vis='15A-305.ms',
        field='1',
        gaintable=['15A-305.antpos', '15A-305.fluxscale1',
                  '15A-305.K0', '15A-305.B0'],
        gainfield=['', '1', '', ''],
        interp=['', 'nearest', '', ''],
        calwt=False)

applycal(vis='15A-305.ms',
        field='2',
        gaintable=['15A-305.antpos', '15A-305.fluxscale1',
                  '15A-305.K0', '15A-305.B0'],
        gainfield=['', '1', '', ''],
        interp=['', 'linear', '', ''],
        calwt=False)
```

Calibration tables that need to be applied are specified by the *gaintable* parameter. If the table has more than one source, the field must be written in the *gainfield*. In the *interp* apply the 'nearest' solution to the calibrators and 'linear' solution to the target field. Parameter *calwt* uses system calibrations to weight the data but this doesn't work for the VLA data yet.

Now we have to examine the newly-calibrated data (the corrected column):

```
plotms(vis='15A-305.ms', field='0', correlation='RR,LL', antenna='',
       avgtime='60', xaxis='channel', yaxis='amp', ydatacolumn=
       'corrected', coloraxis='corr', plotfile=
       'plotms_15A-305-fld0_corrected_amp.png')

plotms(vis='15A-305.ms', field='0', correlation='RR,LL', antenna='',
       avgtime='60', xaxis='channel', yaxis='phase', ydatacolumn=
       'corrected', coloraxis='corr', plotrange=[-1,-1,-180,180],
       plotfile='plotms_15A-305-fld0_corrected_phase.png')
```

The same plots should be examined for the *field*='1', too.  
Final plot to make is 'amp vs UVdist' again.

```
plotms(vis='15A-305.ms', yaxis='amp', xaxis='UVdist', coloraxis=
       'baseline')
```

If there are no problems in the plot, we consider the calibration to be done.

Now we can split off our science target data into separate MS file containing only the field number 2. This call will copy the visibilities from the CORRECTED\_DATA column of the calibrated .ms file and create a new .ms file, where these visibilities will be written to the DATA column. The command used for this step is called *split*:

```
split(vis='15A-305.ms', outputvis='ngc2329.ms', datacolumn=
      'corrected', field='2', correlation='RR,LL')
```

The *outputvis*='ngc2329.ms' will be the name of the new measurement set that *split* will create. To get the calibrated data we wrote into the old MS using *applycal* only, the *datacolumn*='corrected' have to be specified, otherwise the uncalibrated column will be split off. The field containing the desired target has ID number '2' (check in the *listobs* output).

To correct the data weights (the weight and sigma columns) we need to run the *statwt* task. It will remove the effects of relative noise scatter that may have been introduced from flagging.

```
statwt(vis='ngc2329.ms', datacolumn='data')
```

Because splitting can cause new features to appear, we need to check whether there are any, and eliminate them if present. Let's check the 'amp vs UVwave' plot (for later use the 'UVwave' is now more convenient than 'UVdist'):

```
plotms(vis='ngc2329.ms', xaxis='uvwave', yaxis='amp', ydatacolumn=
      'data', field='0', avgtime='30', correlation='RR', coloraxis=
      'baseline')
```

We see that there are some new high peaks, so we will flag them. By marking the region we find out that spectral windows '1', '2' and '4' are corrupted, so we will flag them:

```
flagdata(vis='ngc2329.ms', mode='manual', flagbackup=True, spw=
        '1, 2, 4')
```

Check the plot once again. This time it should be much better and we can start imaging.

## 4.4 Imaging

Firstly we start with so-called dirty image. We will not further work with this image but we will use it later for comparison. For self-calibration and creating and improving the image we use the task `tclean`.

```
tclean(vis='ngc2329.ms', imagename='images/dirty',
       datacolumn='data',
       niter=0,
       imsize=[8000], cell=['0.07arcsec'],
       weighting='briggs', robust=0.0,
       savemodel='none')
```

To keep the working directory clear, we can create a new sub-directory called 'images' before running `tclean`. In this directory, all of the images will be stored. Then we design the path to that directory in the *imagename* parameter and after the last slash we write down the name of the new image. Since we are doing just the dirty image now, we do not need to open it in the interactive environment, that is why the *niter* is equal to '0'.

Before running `tclean` we need to calculate the image size and the cell size.

Ideally the image should include all the sources present in the data, so the image should be large enough. The equation for the VLA primary beam size is set to  $42 \cdot 1/\nu$ , where  $\nu$  is in GHz. At the center frequency of the L-band (1.5 GHz) the VLA primary beam has FOV of 28 arcmin. To image the entire primary beam and the first sidelobe, we need the image field of view about two times larger.

From the [Resolution tables \(2021\)](#) we found out that the resolution for the L-band in the A-configuration of the VLA is about 1.3 arcsec. Recommended value of pixels across the FWHM of the synthesized beam is 5 but in our case that was not enough (the center of the emission was barely recognizable, so we chose three times more the pixels). That gives us the cell size of 0.07 arc seconds.

Now we convert the field of view to pixels using the cell size. A formula to calculate image size in pixels for our case is following:  $28' \cdot (60''/1.3') \cdot (1\text{pixel}/0.20'')$ . The result gives us image size of about 12 000 pixels (for later inspection we will use the image size of only 8 000 pixels).

Next important parameter is the *weighting*. The most used options are the 'natural', 'uniform' and 'briggs'. The *natural* weighting is used when the maximum imaging intensity is needed at the expense of a possibly thick PSF. On the other hand, *uniform* weighting is the most appropriate to experiments where a well behaved PSF is important. The PSF has then a narrow main lobe and suppressed sidelobes. For our experiment we use the *briggs* weighting which lies between the natural and uniform. This option has the additive parameter named *robust* with which we can tell the `tclean` task if we want more of



the *natural* or the *uniform* properties. The *robust* parameter can vary from  $-2.0$  to  $+2.0$ , where the negative means 'uniform' and the positive means 'natural'. For our purposes we want to have balanced ratio between the PSF and the intensity of the image, so we chose *robust*=0.0 (Briggs 1995).

The final parameter is the *savemodel*. Because this is just the dirty image and we will not make any model, we don't need to save anything. That is why we chose this parameter to be equal to 'none'.

A self-calibration is used to improve the image of the source because it can compensate some of the calibration errors and eliminate extensive artifacts from potential background source.

It is divided into several rounds. Each round consists of cleaning, creating the calibration table, inspecting solutions, and applying to MS.

The first round starts with creating an initial model. The task *tclean* is being used again. This time we need to set the *interactive*=True so we will be able to create the mask and control the cleaning depth that is essential to cleaning.

```
tclean(vis='ngc2329.ms', imagename='images/ngc2329_I',
       imsize=[8000], cell=['0.07arcsec'],
       deconvolver='mtmfs', nterms=2,
       weighting = 'briggs', robust=0.0,
       niter=100000,
       interactive=True,
       savemodel='modelcolumn')
```

Image size and cell size are computed the same way as in the previous step and so is the weighting. To produce the model of the field, we need to deconvolve the image. As our image combines data from all of the spectral windows, the source's amplitude may vary over the frequency range and may cause errors. To compensate this effect, we set the *deconvolver*='mtmfs' with subparameter *nterms*=2 that will create spectral index map of the source at the end of imaging.

The cleaning takes a few cycles to be done and number of these cycles is set by the *niter* parameter. We normally do not want to be limited by the cycles left so we choose some higher number that is nearly impossible to reach. For our purposes, 100 000 is just fine.

To let the model be written to the MODEL\_COLUMN after imaging, we have to change the default setting of the *savemodel*='none' to *savemode*='modelcolumn'. Everything is now ready to run the *tclean*.

In the interactive window that will show up after a while we need to zoom the central part to see the central emission of the source. In the mouse toolbar (third row from the top) click on the 'R' letter in the oval and start marking the brightest emission (red to white color). After the mask is set, confirm the mask and run the first cycle of the process by clicking on the green twisted arrow in the green box. It takes few minutes to complete the cycle. For the initial mask it is enough to run this cycle just once. After the first cycle is done, click on the red sign with cross in the 'Next action' section in the green box.

After finding applicable *solint*, we can run *gaincal* task.

```
gaincal(vis='ngc2329.ms', caltable='ngc2329.selfcal1', solint='6s',
        refant='ea09', calmode='p', gaintype='T', minsnr=3)
```

The same conditions as for the *refant* selection for the calibration are also applicable for `tclean`. *calmode*='p' means only the phase correction will be calculated. Another option is 'a' for amplitude correction and 'ap' for both the amplitude and the phase. The *gaintype*='T' obtains one solution for both polarizations. The final parameter, *minsnr*, flags all the points for which the signal-to-noise ratio is smaller than 3.

After generating this table we examine its values:

```
plotms(vis='ngc2329.selfcal1',
        xaxis='time',
        yaxis='phase',
        iteraxis='antenna')
```

If satisfied with the solutions, apply the calibration to the MS.

```
applycal(vis='ngc2329.ms',
          gaintable=['ngc2329.selfcal1'], gainfield=[''],
          interp=['nearest'],
          calwt=[False],
          applymode='calflag')
```

Now we can start a new round of self-calibration. To save the time spend on each cleaning, we can reuse the mask created in the last round and add it to the call for `tclean`. To do this, just add the parameter *mask* to the call and write down the path to the latest mask created (the input expected is again the string). This time we can clean deeper than in the first round (e.g., about three cycles should work).

```
tclean(vis='ngc2329.ms', imagename='images/ngc2329_II',
        imsize=[8000], cell=['0.07arcsec'],
        deconvolver='mtmfs', nterms=2,
        weighting = 'briggs', robust=0.0,
        niter=100000,
        interactive=True,
        mask='images/ngc_2329.mask',
        savemodel='modelcolumn')
```

Generate the calibration table again and apply it to the MS. Don't forget to include previously created calibration table.

```

gaincal(vis='ngc2329.ms', caltable='ngc2329.selfcal2',
        gaintable='ngc2329.selfcal1',
        solint='6s',
        refant='ea09',
        minsnr=3,
        gaintype='G', calmode='p')

applycal(vis='ngc2329.ms',
         gaintable=['ngc2329.selfcal1', 'ngc2329.selfcal2'], gainfield=
         [''], interp=['nearest'],
         calwt=[False],
         applymode='calflag')

```

For the third cycle start with the mask created in the second run and so on. Repeat these rounds until the RMS stops decreasing. To determine RMS type viewer in the CASA terminal, find images you need to inspect, and open them as raster image. Mark region that does not include any emission and pinpoint the RMS field from the statistics (figure 4.7).

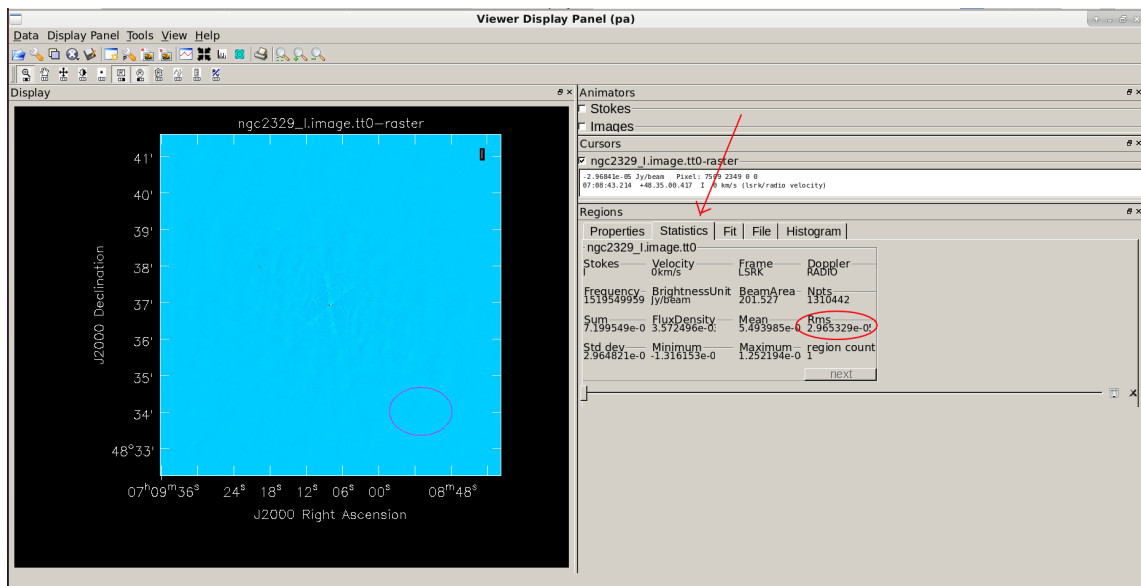


Figure 4.7: This figure shows where to find the RMS in the viewer

When the RMS stops decreasing, the self-calibration is done. Then run the `tclean` again with parameter `pbcor=True` and clean as deep as possible. The final image of this thesis can be seen in the Result section.

```

tclean(vis='ngc2329.ms', imagename='images/ngc2329_final',
       imsize=[8000], cell=['0.07arcsec'],
       deconvolver='mtmfs', nterms=2,
       weighting = 'briggs', robust=0.0,
       niter=100000,
       interactive=True,
       pbcor=True,
       savemodel='modelcolumn')

```

## 4.5 Spectral index map

Since spectral index shows the dependence of the flux density on frequency, we need to download another image of this source but in another band. From the [Image archive \(2021\)](#) we can download the desired image by filling up the 'Object name' and the coordinates of the source.

For this thesis we chose the C-band observation from the year 1995 and then we downloaded FITS image and the calibrated u-v data. Again, after some time an email arrived.

First thing is to create the MS from the file we downloaded:

```

importuvfits(fitsfile='4.74F27BB_AC445_1995DEC22_1.uvfits',
             vis='ngc2329_Cband_Bconfig.ms')

```

For creating MS from the *.uvfits* files we use the task `importuvfits`, where the *fitsfile* is the original file.

For later calculations we need to know what is this observation's longest baseline in the wavelengths, so we plot the 'amp vs UVwave' and save the output.

```

plotms(vis='ngc2329_Bband_Bconfig.ms', xaxis='uvwave', yaxis='amp',
       plotfile='plotms_ngc2329_Cband_Bconfig_amp_vs_uvwave.png',
       coloraxis='baseline')

```

To make spectral index map we also need an image: the second file downloaded has the format of *.imfits* but it is not suitable for using in CASA. We need to convert it to the CASA-friendly format.

```

importfits(fitsimage='4.74I1.48_AC445_1995DEC22_1_158.Ua2.73M.imfits',
           imagename='../spectral_index/ngc2329_Cband_Bconfig.image')

```

Save the image to a new dictionary named 'spectral\_index'. It must be on the same level as the current dictionary.

For the map we need to compensate the different array configurations. That is why we need to clean the L-band image again with new parameters *uvtaper* and *uvrange*.

By opening the viewer in CASA and choosing the 'ngc2329\_Cband\_Bconfig.image' in the data manager we see the restoring beam information in the right loading options column. We do not need to open the image, this information is enough. To the *uvtaper*

parameter we write down the numbers we just determined. The next parameter is the *uvrange*. There comes the longest baseline of the array in wavelength for the C-band image.

```
#in the L-band working dictionary

tclean(vis='ngc2329.ms',
        imagename='../spectral_index/ngc2329_Lband_specind',
        imsize=[8000], cell=['0.07arcsec'],
        deconvolver='mtmfs', nterms=2,
        weighting = 'briggs', robust=0.0,
        niter=100000,
        interactive=True,
        pbcor=True,
        uvrange='0~180klambda',
        uvtaper=['1.647arcsec', '1.325arcsec', '80.85deg'],
        mask='images/ngc2329_final.mask',
        savemodel='modelcolumn')
```

After cleaning, we need to regrid the L-band image to have the same image size as the C-band image. To do this, we use the *imregrid* task.

```
#in spectral index dictionary

imregrid(imagename='ngc2329_Lband_specind.image.tt0', template=
        '../ngc2329_historical_Cband_Bconfig/ngc2329_Cband_Bconfig.
        image/', shape=[957,957,1,1], output=
        'ngc2329_Lband_specind_regrid.image')
```

Now we have two images with the same image size. Final step before creating the map is to cut off the spectral and Stokes axes, since they are incompatible for both the images.

```
imsubimage(imagename='ngc2329_Lband_specind_regrid.image', outfile=
        'ngc2329_Lband_specind_regrid_subim.image', stokes='I',
        dropdeg=True)

imsubimage(imagename='ngc2329_Cband_Bconfig.image', outfile=
        'ngc2329_Cband_Bconfig_specind_regrid_subim.image', stokes='I',
        dropdeg=True)
```

Now that everything is set, we can move to creating the map.

For two given frequencies of two different images with flux densities we can measure the spectral index (which is the slope of the power law shape of the spectrum) from the relation

$$\alpha = \frac{\log \frac{S_{\nu_2}}{S_{\nu_1}}}{\log \frac{\nu_2}{\nu_1}}, \quad (4.2)$$

where  $S_{v_1}$  and  $S_{v_2}$  are the flux densities of the images and the  $v_1$  and  $v_2$  are the frequencies.

From both the images, we determine the threshold ( $3 \times \text{RMS}$ ) and the observed frequency:

```
freq_Lband='1.5120'
freq_Cband='4.7101'
threshold_Lband='9.398133e-5'
threshold_Cband='5.497428e-4'
```

To calculate the spectral index and create the map we use the task `immath`.

In this parameter, we need to use the region: in the viewer open the L-band image and with the 'R' in the polygon outline the whole emission. Then in the 'Open Data' button select the 'save region'. We named our region 'specind\_region\_I'.

```
tget(immath)
imagename=['ngc2329_Lband_specind_regrid_subim.image',
'ngc2329_Cband_Bconfig_specind_regrid_subim.image']
mode='evalexpr'
expr='(log(IM1[IM1>'+ threshold_Cband+']/IM0[IM0>'+threshold_Lband+
']))/log((''+freq_Cband+'/' +freq_Lband+'))'
outfile='ngc2329_LC_spim.image'
region='specind_region_I'
```

In the *imagename* parameter it is important to have the image of lower frequency first. *mode='evalexpr'* is used when we want to input equations and the *expr* is the parameter where the equation is written. The final image will be stored under the name 'ngc2329\_LC\_spim.image'.

If everything is set correctly, type 'go()' and run the task. The resulting image will occur in the directory.

# Chapter 5

## Results

After several cleaning rounds (see chapter 4), the final image emerges. In CASA viewer we can open the 'ngc2329\_final.image.tt0' as a raster image.

To see the whole emission, we need to set the minimum of the data range as the threshold ( $5 \times \text{RMS}$ ) and change the scaling power cycles to a lower value until the emission is nicely visible.

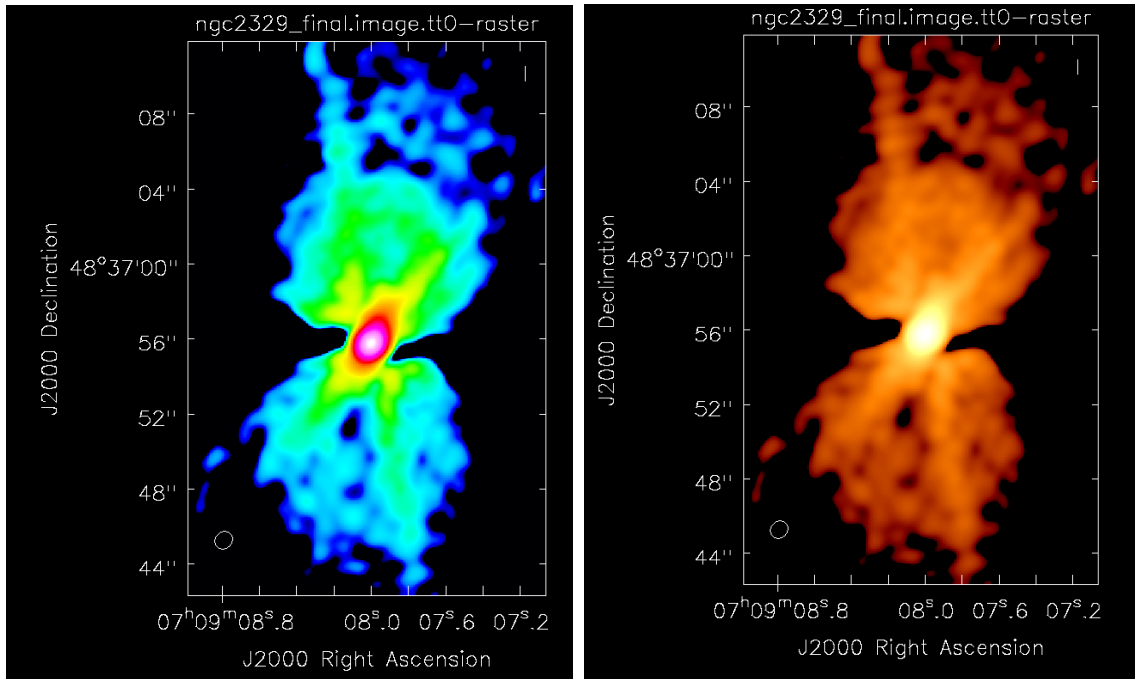


Figure 5.1: Final image of the emission from the NGC2329 in Rainbow 2 color map (*left*) and in Hot Metal 2 color map (*right*).

From both the images we can see the lobes of the galaxy but the asterisk shape caused by the interferometer is strongly visible, too. Unfortunately, we were not able to reduce this pattern. The reason is probably badly chosen secondary calibrator during the observation. Software CASA only works with the point-source calibrators, but this particular calibrator was resolved, thus there were many RFIs left.

Also, bigger image size could have been chosen for the self-calibration so more external sources could have been included.

Next we have the spectral index map. From the left image in the figure 5.2 we see that in the center the spectral index varies between  $-0.5$  and  $0$ , so the spectrum is flat. That refers to continuous injection of new electrons from the core, while on the edges the spectral index reaches values higher than  $1$ , so the spectrum becomes steeper and steeper.

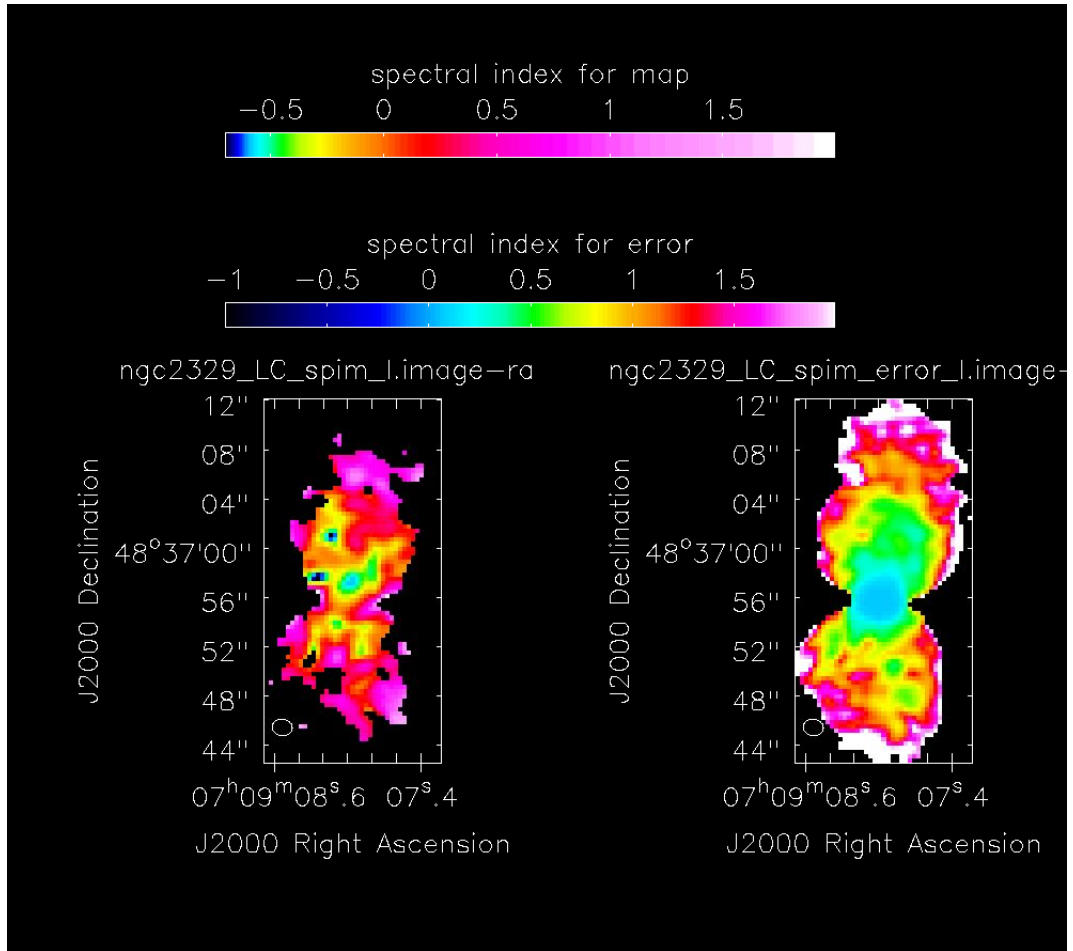


Figure 5.2: Final spectral index map (*left*) and the spectral index map error (*right*). The lowest error occurs in the core, where the map is the most accurate.



# Conclusion

Purpose of this thesis was to get knowledge of the different types of radio galaxies and learn to work with CASA software. We calibrated observed radio data from the VLA observatory, made image of the galaxy NGC 2329, and created the spectral index and spectral index error map of that source.

In the introductory part, we gave some brief elementary information about the electromagnetic spectrum and the synchrotron radiation (chapter 1), the radio interferometry (chapter 2) and radio galaxies (chapter 3). In the second, processing part (chapter 4), we described the data reduction and imaging process on this specific target.

At the end of the imaging, unfortunately, we found out that the image still contains the interferometric asterisk shape (see in the figure 5.1) that might be caused by the resolved secondary calibrator. The final image could be improved if new model (the real emission area) of the extended calibrator was included in the calibration, and perhaps bigger image size could be used during the self-calibration. At the end, we created the spectral index map with the error map (figure 5.2) which nicely shows the 'aging' of the electrons injected by the core.

This thesis could be used by the students as a guide for CASA software as each command is described in detail. The source NGC 2329 is not well explored yet, so I would like to inspire anyone interested in radio astronomy to re-process the data and get more precise results possibly with the suggestions written above.



# Bibliography

- Bridle, A. H., Hough, D. H., Lonsdale, C. J., Burns, J. O. & Laing, R. A. (1994), ‘Deep VLA Imaging of Twelve Extended 3CR Quasars’, **108**, 766.
- Briggs, D. S. (1995), *High fidelity deconvolution of moderately resolved sources*, Socorro, New Mexico.
- Ertley, C. (2014), ‘Studying the polarization of hard x-ray solar flares with the gamma ray polarimeter experiment (grape)’.
- Fanaroff, B. L. & Riley, J. M. (1974), ‘The morphology of extragalactic radio sources of high and low luminosity’, **167**, 31P–36P.
- Feretti, L., Giovannini, G., Gregorini, L., Padrielli, L., Roland, J. & Valentijn, E. (1985), ‘The wide angle tailed radio source ngc 2329 in the cluster a 569’, *Astronomy and Astrophysics* **147**, 321–327.
- Image archive* (2021). [Accessed: 31.5.2021].  
**URL:** <https://archive.nrao.edu/archive/archiveimage.html>
- Jansky, K. G. (1933), ‘Electrical disturbances apparently of extraterrestrial origin’, *Proceedings of the Institute of Radio Engineers* **21**(10), 1387–1398.
- NRAO (2021), ‘VLA continuum tutorial’. [31.05.2021].  
**URL:** [https://casaguides.nrao.edu/index.php?title=VLA\\_Continuum\\_Tutorial\\_3C391-CASA5.7.0](https://casaguides.nrao.edu/index.php?title=VLA_Continuum_Tutorial_3C391-CASA5.7.0)
- NRAO data archive* (2020). [Accessed: 17.11.2020].  
**URL:** <https://archive.nrao.edu/archive/advquery.jsp>
- Operator’s log* (2020). [Accessed: 17.11.2020].  
**URL:** <http://www.vla.nrao.edu/cgi-bin/oplogs.cgi>
- Perley, R. A. & Butler, B. J. (2017), ‘An accurate flux density scale from 50 mhz to 50 ghz’, *The Astrophysical Journal Supplement Series* **230**(1), 7.  
**URL:** <http://dx.doi.org/10.3847/1538-4365/aa6df9>
- Perley, R. A., Willis, A. G. & Scott, J. S. (1979), ‘The structure of the radio jets in 3C 449.’, **281**, 437–442.

Rayleigh (1879), ‘XXXI. Investigations in optics, with special reference to the spectro-scope’.

**URL:** <https://doi.org/10.1080/14786447908639684>

*Resolution tables* (2021). [Accessed: 30.5.2021].

**URL:** <https://science.nrao.edu/facilities/vla/docs/manuals/oss2017B/performance/resolution>

Saikia, D., Konar, C., Jamrozy, M., Machalski, J., Gupta, N., Stawarz, L., Mack, K.-H. & Siemiginowska, A. (2007), ‘Episodic activity in radio galaxies’.

Shapley, H. (1918), ‘Studies based on the colors and magnitudes in stellar clusters. VII. The distances, distribution in space, and dimensions of 69 globular clusters.’, **48**, 154–181.

Wiedemann, H. (2019), *Particle Accelerator Physics*, Springer International Publishing.

Wille, K. (2013), ‘Synchrotron radiation lecture notes’, p. 65.